To Copy or Not to Copy: Making In-Memory Databases Fast on Modern NICs

THE UNIVERSITY OF UTAH®

Aniraj Kesavan
Robert Ricci
Ryan Stutsman

Introduction

- Didn’t we solve DB I/O when we got rid of disks ?
- Today: Copy records to transmit buffer to send
- Zero copy: Transmit Data directly from records
- Do you copy or not?
- Is there a way to get best of both worlds?

The Setup

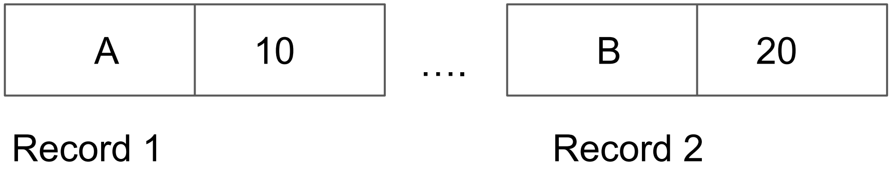
- Mellanox Infiniband Cx-3 and Connect-ib
- Peak B/W - 5.8 GB/s, latency ~ 1 **µ**s, kernel bypass
- 15 clients:
  - Copy Out
  - Zero Copy

Copy Out

A	10

Record 1

Copy Out

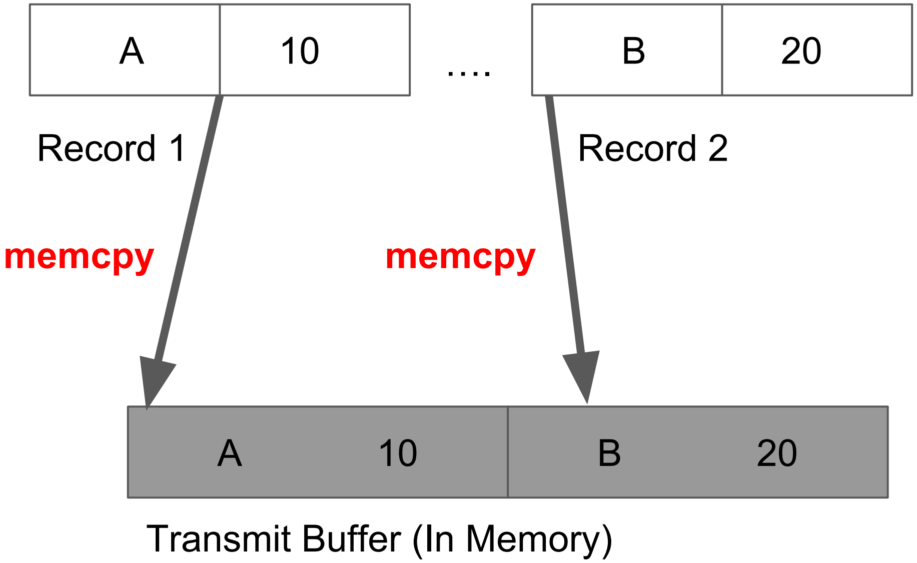


A 10 …. B 20

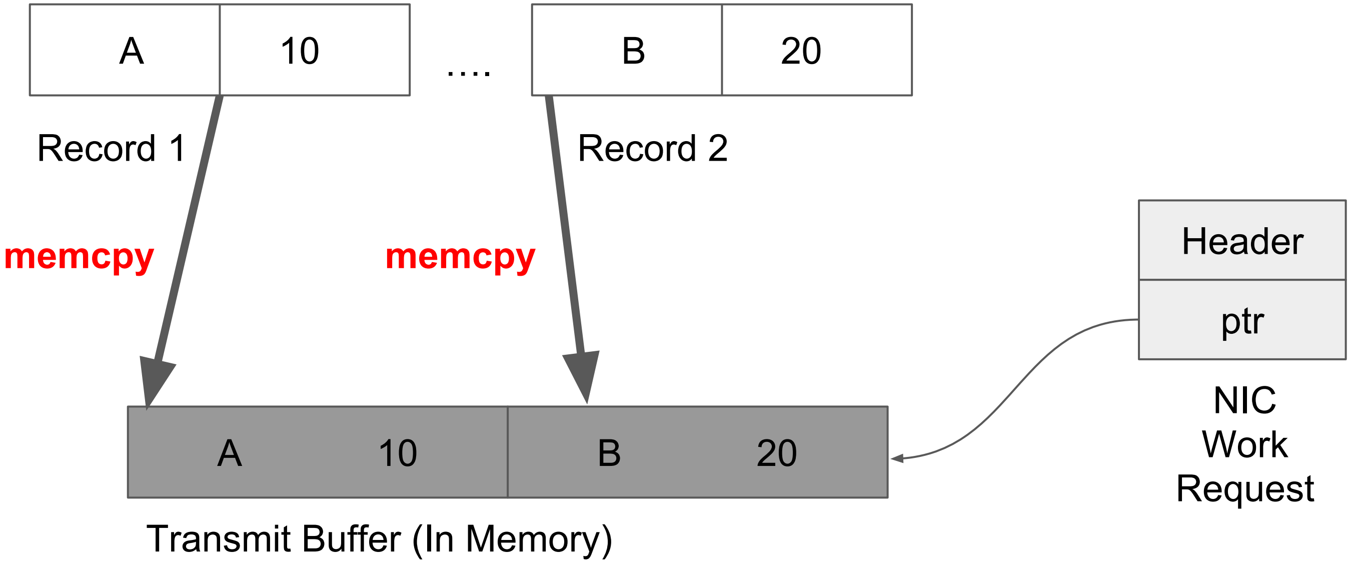
Record 1

Record 2

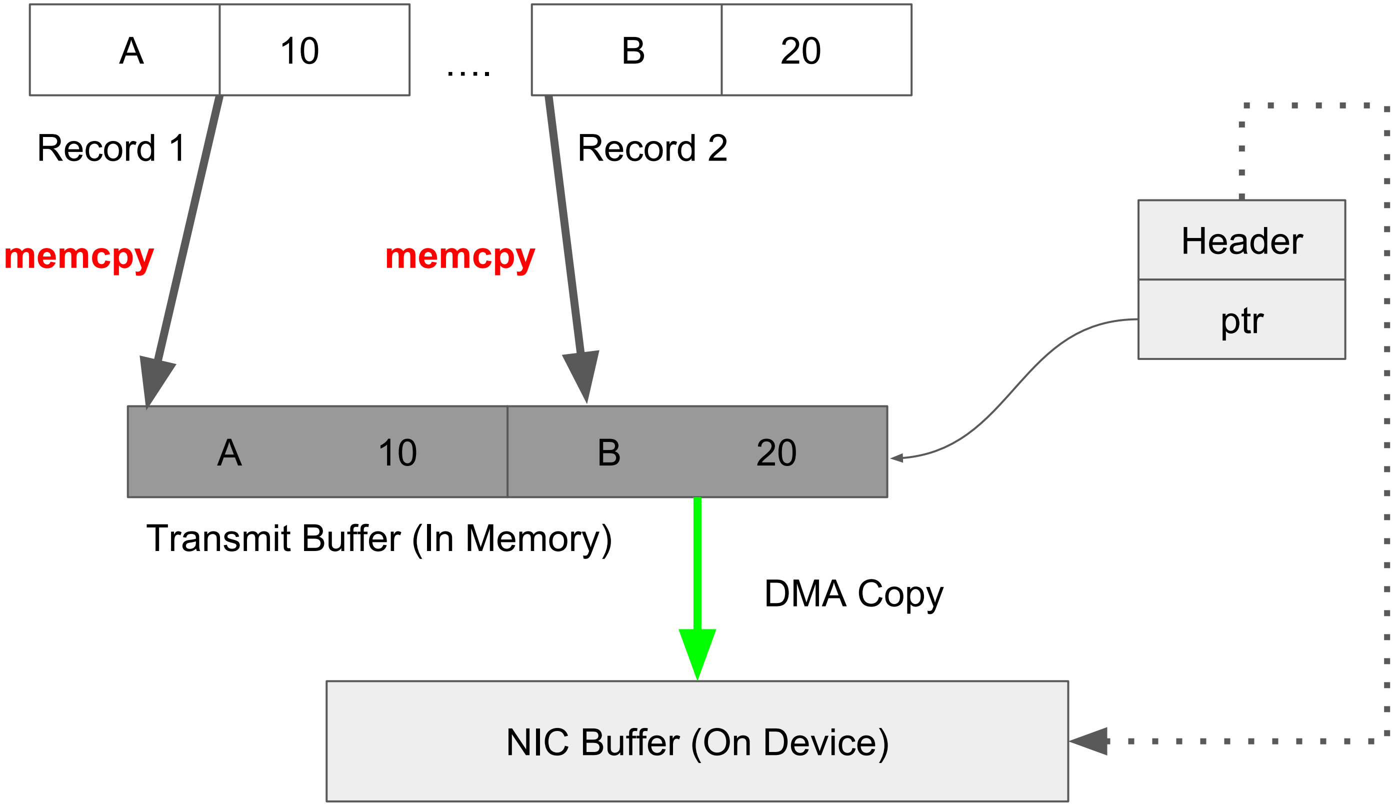
Copy Out


A
10
....
B
20
Record 1
Record 2
memcpy
memcpy
A
10
B
20
Transmit Buffer (In Memory)

Copy Out


A
10
....
B
20
Record 1
Record 2
memcpy
memcpy
Header
ptr
NIC
Work
Request
A
10
B
20
Transmit Buffer (In Memory)

Copy Out



A
10
....
B
20
Record 1
Record 2
memcpy
memcpy
Header
ptr
A
10
B
20
Transmit Buffer (In Memory)
DMA Copy
NIC Buffer (On Device)

Takeaways - Copy Out

- Involves CPU cycles to copy

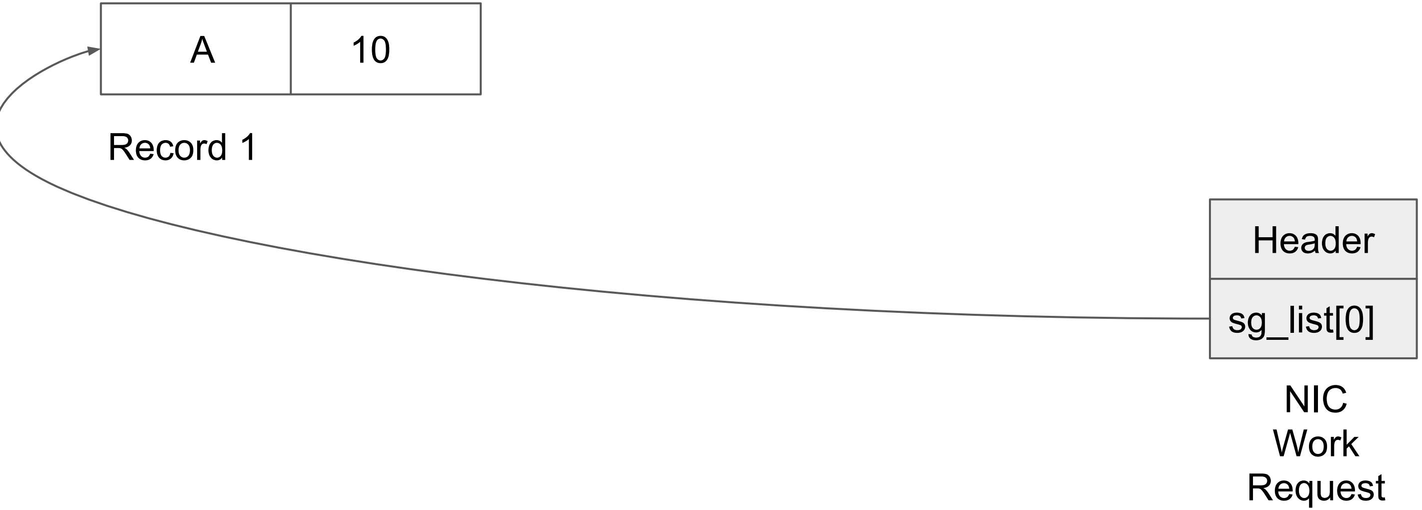
- Involves a memcpy - 2X more memory bandwidth

Zero Copy

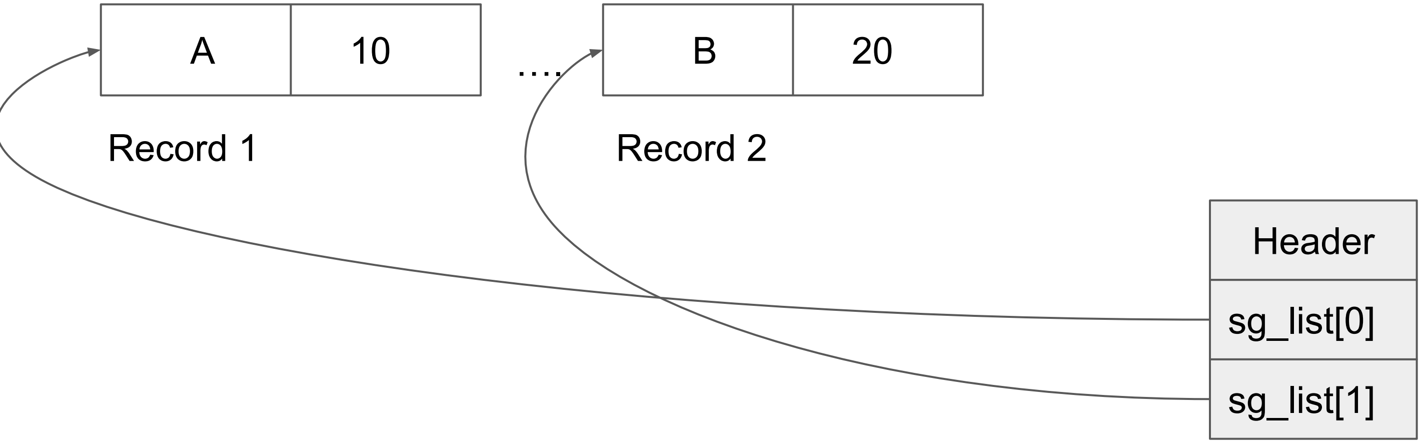
A	10

Record 1

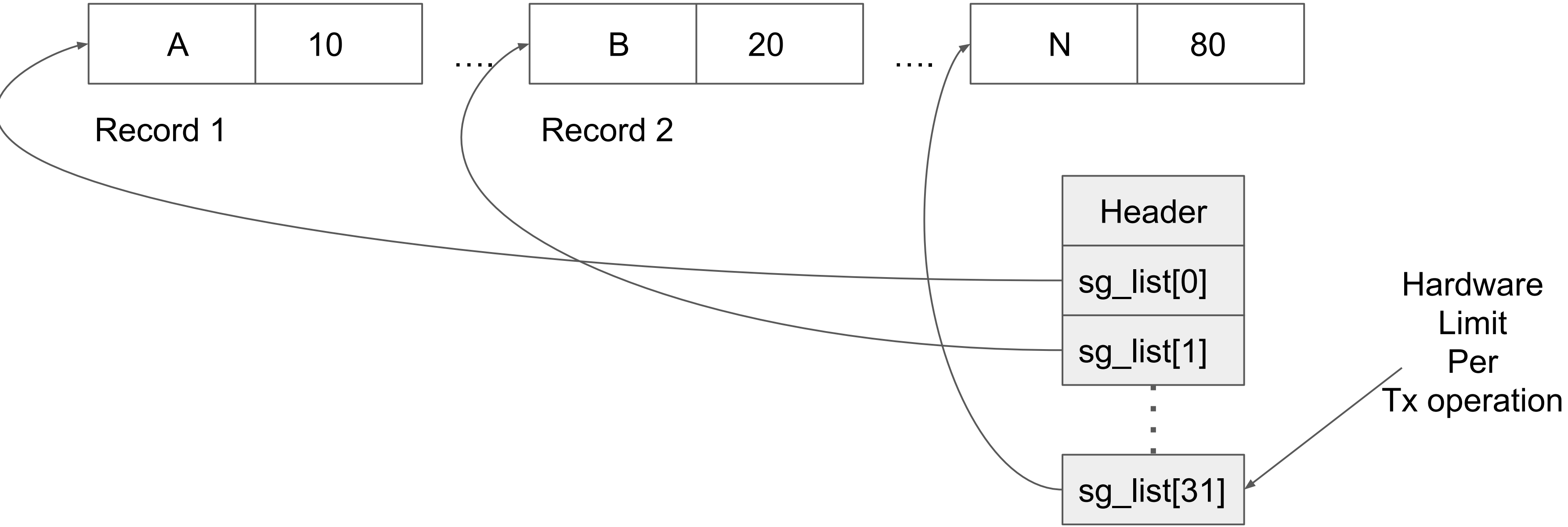
Zero Copy



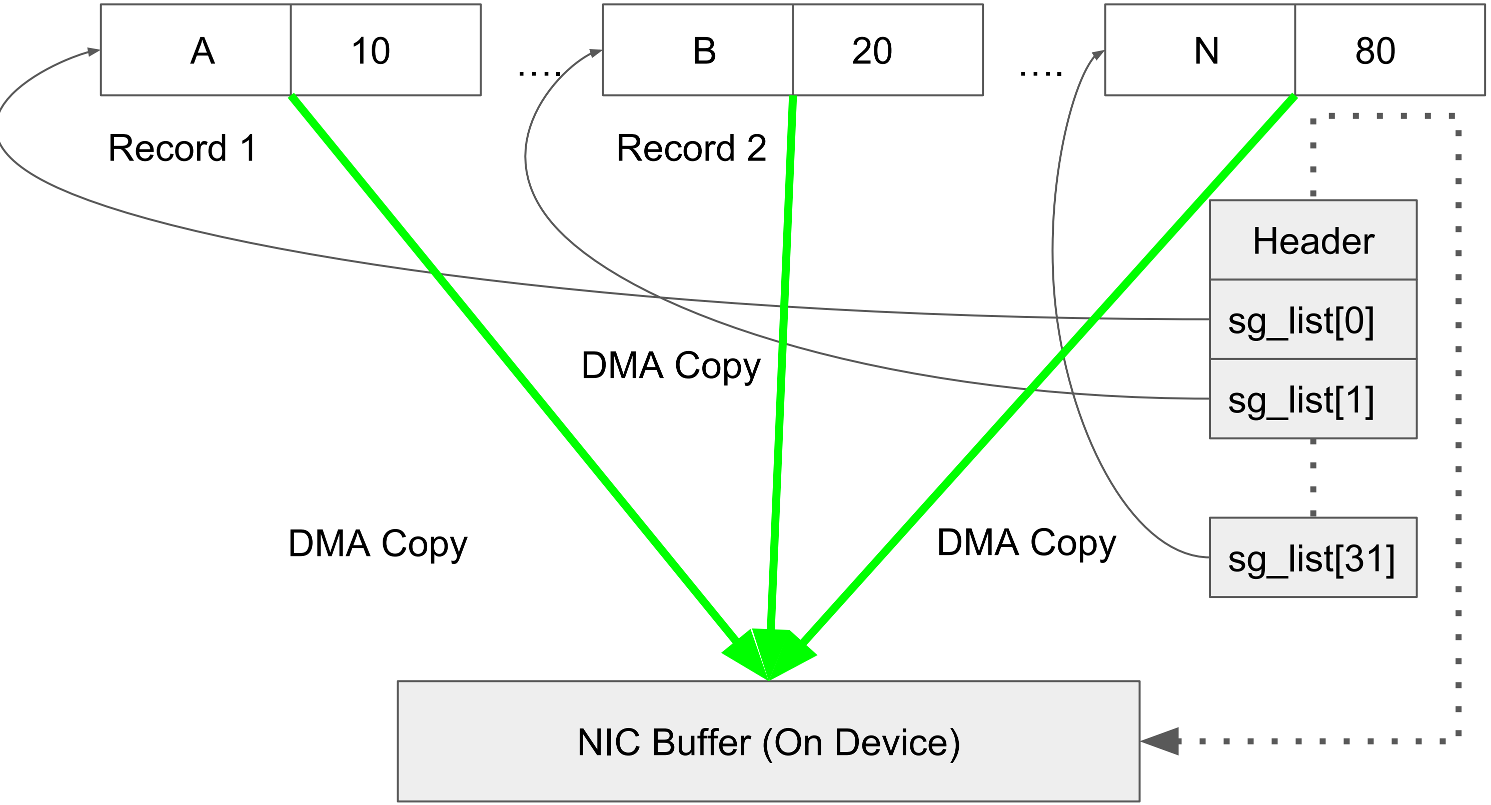
Zero Copy



Zero Copy


A
10
….
B
20
….
N
80
Record 1
Record 2
Header
sg_list[0]
sg_list[1]
sg_list[31]
Hardware
Limit
Per
Tx operation

Zero Copy


A
10
….
B
20
….
N
80
Record 1
Record 2
Header
sg_list[0]
sg_list[1]
sg_list[31]
DMA Copy
DMA Copy
DMA Copy
NIC Buffer (On Device)

Takeaways - Zero Copy

- No memcpy

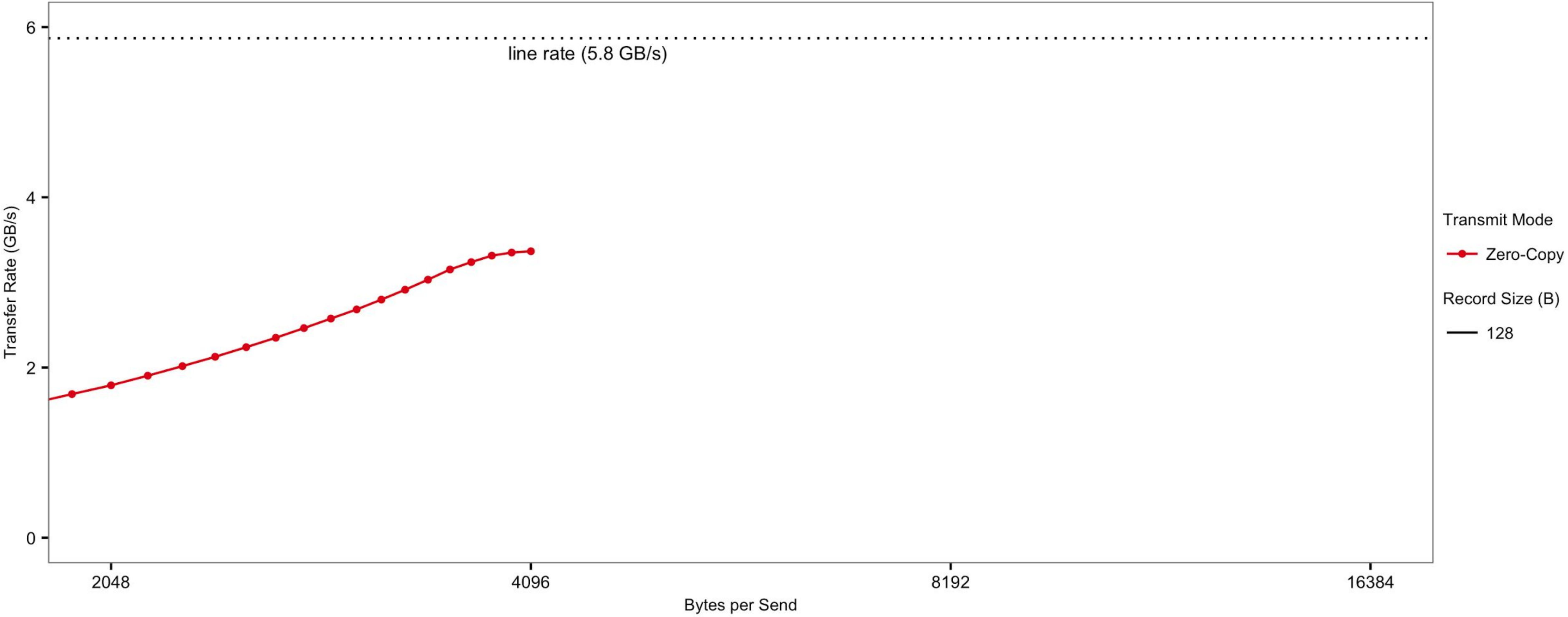
- Limited number of records per transmission

Experiments

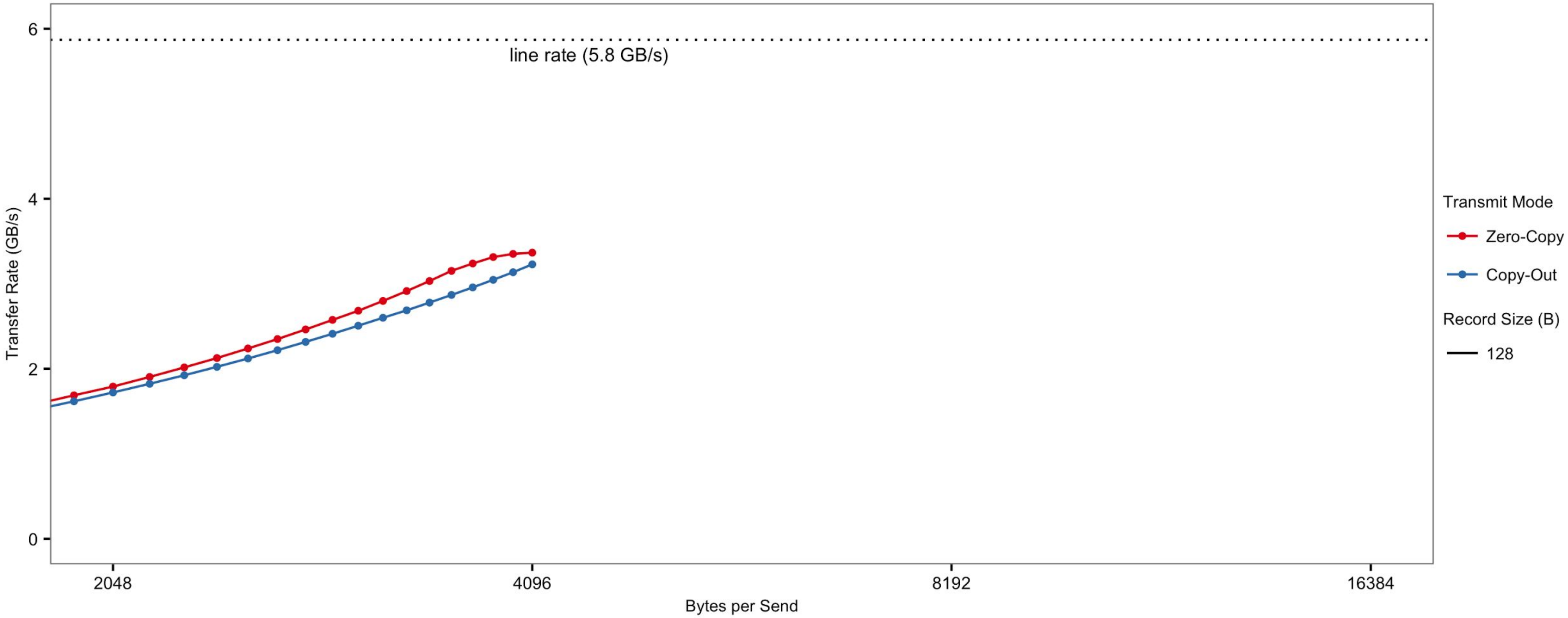
- Measuring effects of layout

- When to use which?

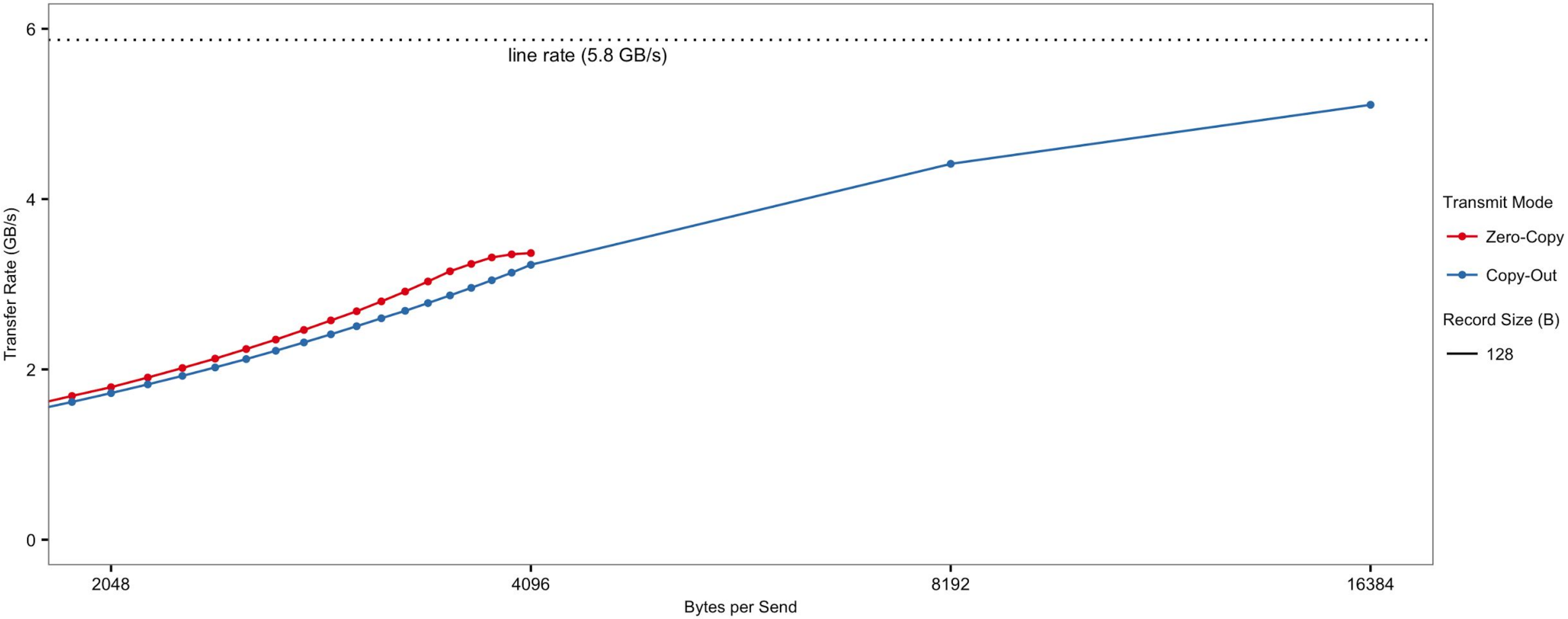
Transmission Throughput (Zero Copy)



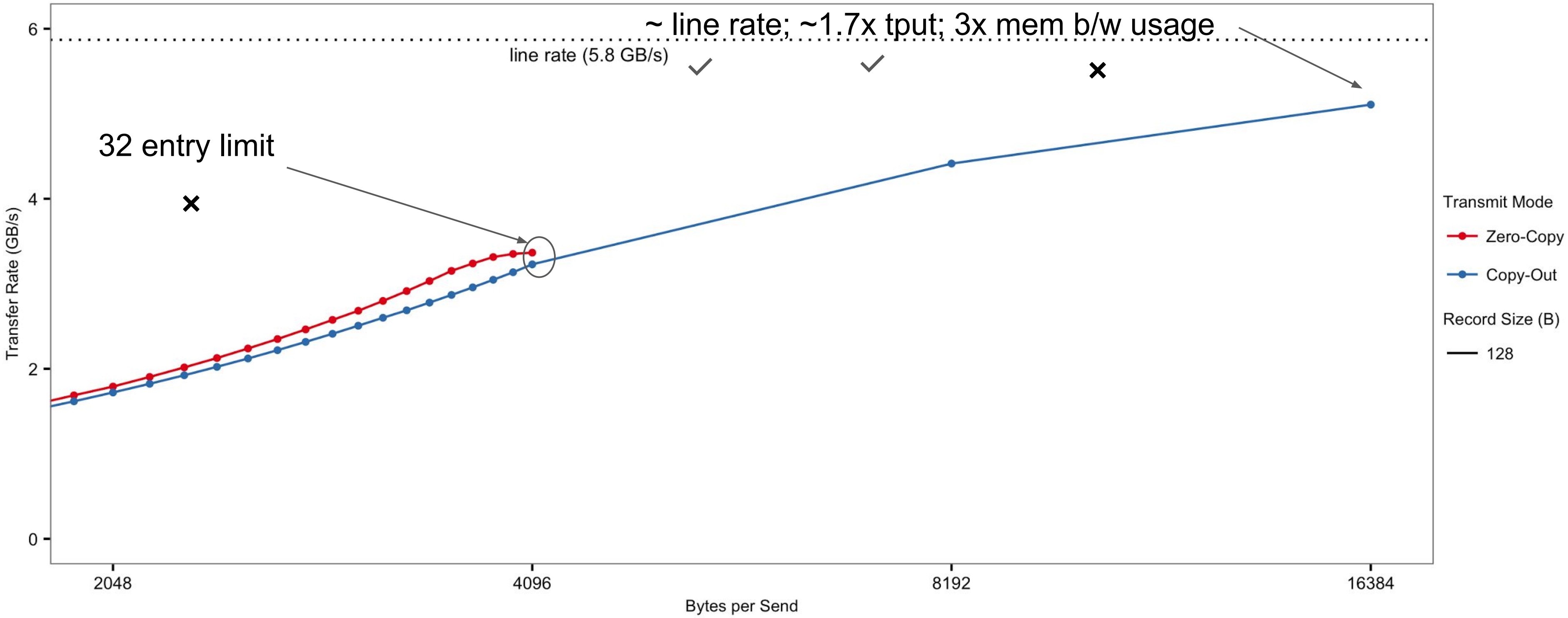
Transmission Throughput (Comparison)



Transmission Throughput (Comparison)



Transmission Throughput (Comparison)



~ line rate; ~1.7x tput; 3x mem b/w usage

32 entry limit

What makes the NIC happy?

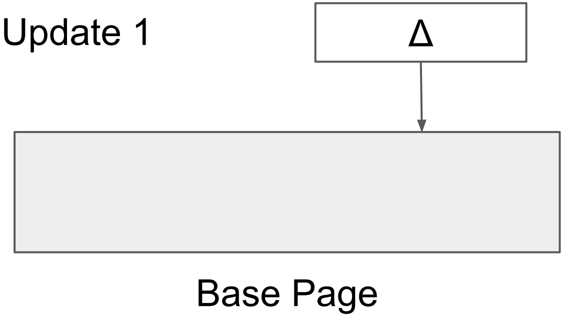
- Large Chunks of data - better throughput
- A few chunks of data that it can gather
- Stable data
    - Zero Copy requires records to be locked over transmission

Bw-Tree

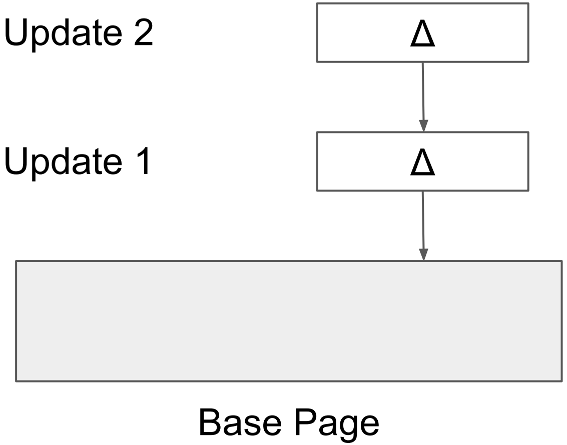
[Bw-Tree - Levandoski et al., 2013]

Base Page

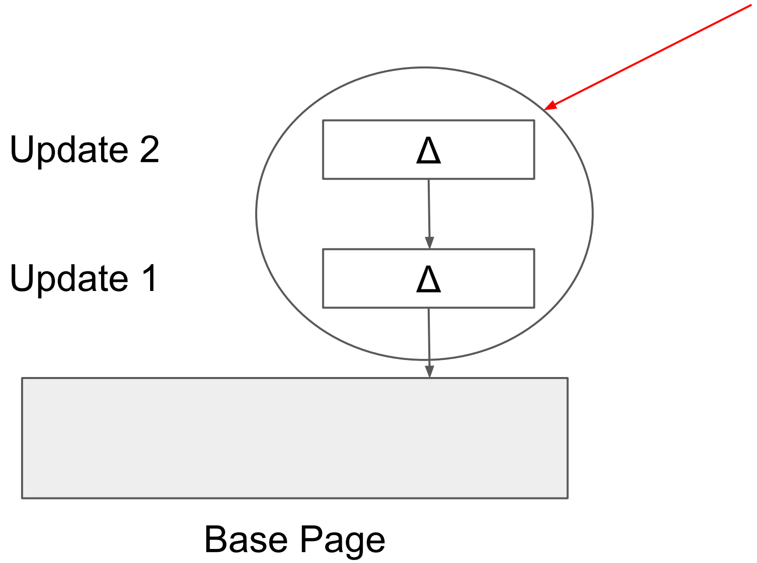
Bw-Tree



Bw-Tree

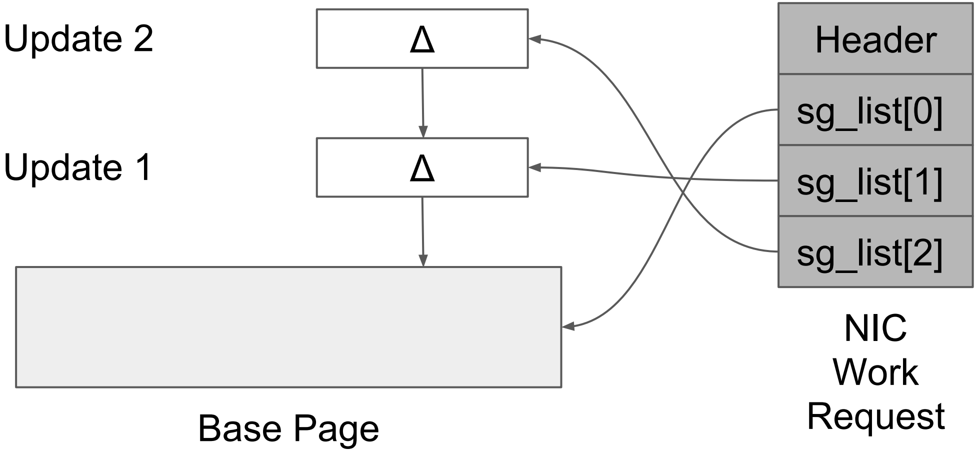


Bw-Tree

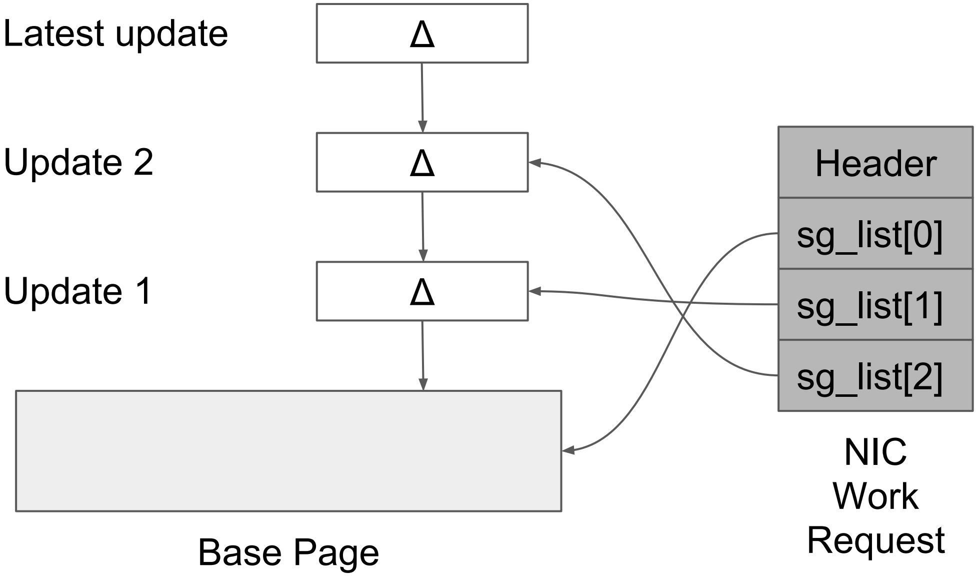


Consolidate to avoid read slowdown?

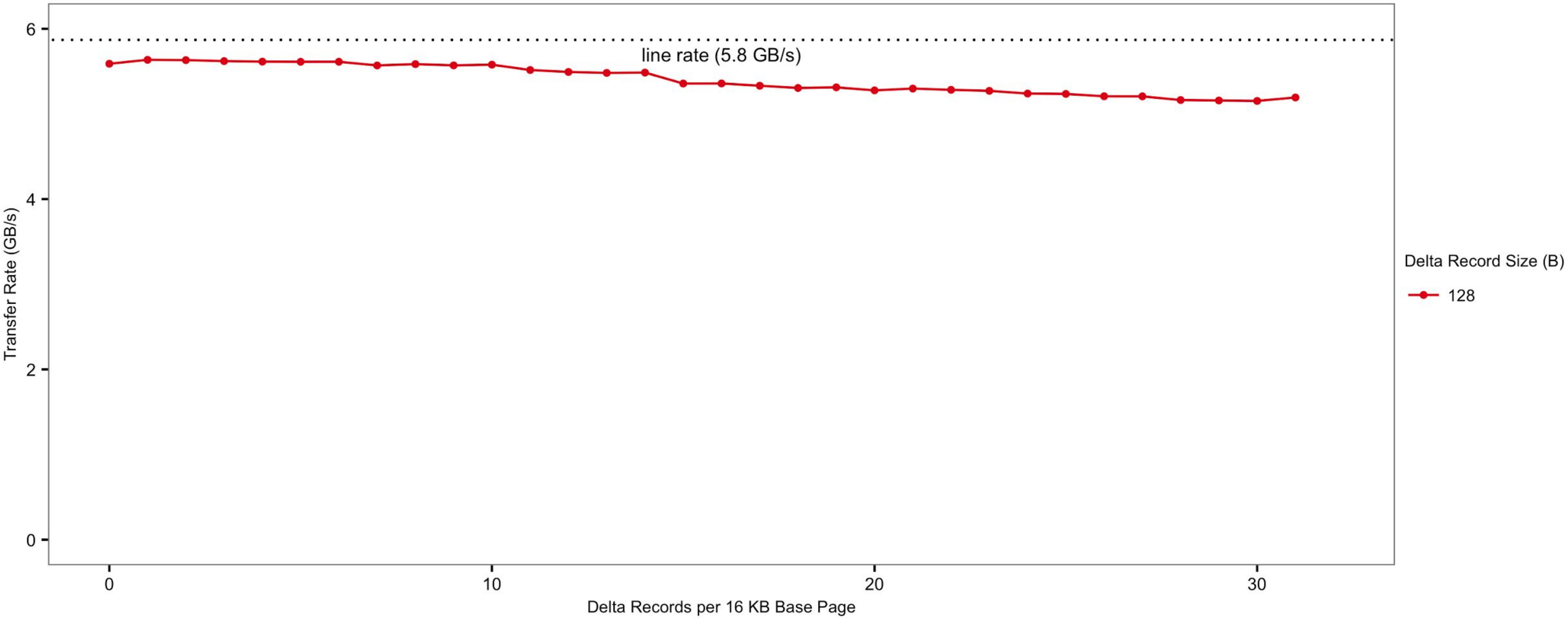
Bw-Tree


Update 2
Δ
Update 1
Δ
Base Page
Header
sg_list[0]
sg_list[1]
sg_list[2]
NIC
Work
Request

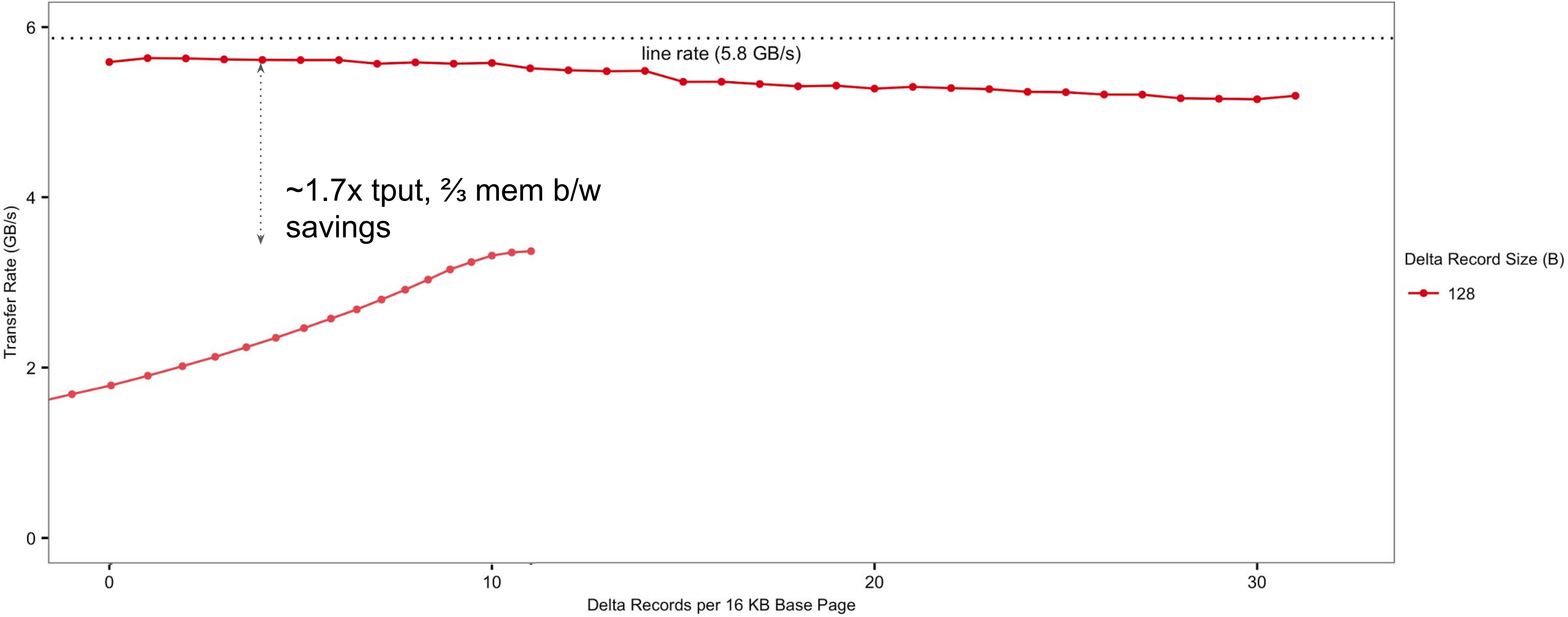
Bw-Tree


Latest update
Δ
Update 2
Δ
Update 1
Δ
Base Page
Header
sg_list[0]
sg_list[1]
sg_list[2]
NIC
Work
Request

Deltas make NIC happy



Deltas make NIC happy



~1.7x tput, ⅔ mem b/w savings

Conclusion

- Does DB layout matter for NIC performance?
  - Yes. If you care about mem b/w and CPU cycles.
- No updates in place structures like Bw-Tree gives us the best of both worlds by:
  - Transmitting bigger chunks directly aiding throughput
  - Transmitting smaller chunks directly saving memory b/w and CPU consolidation costs

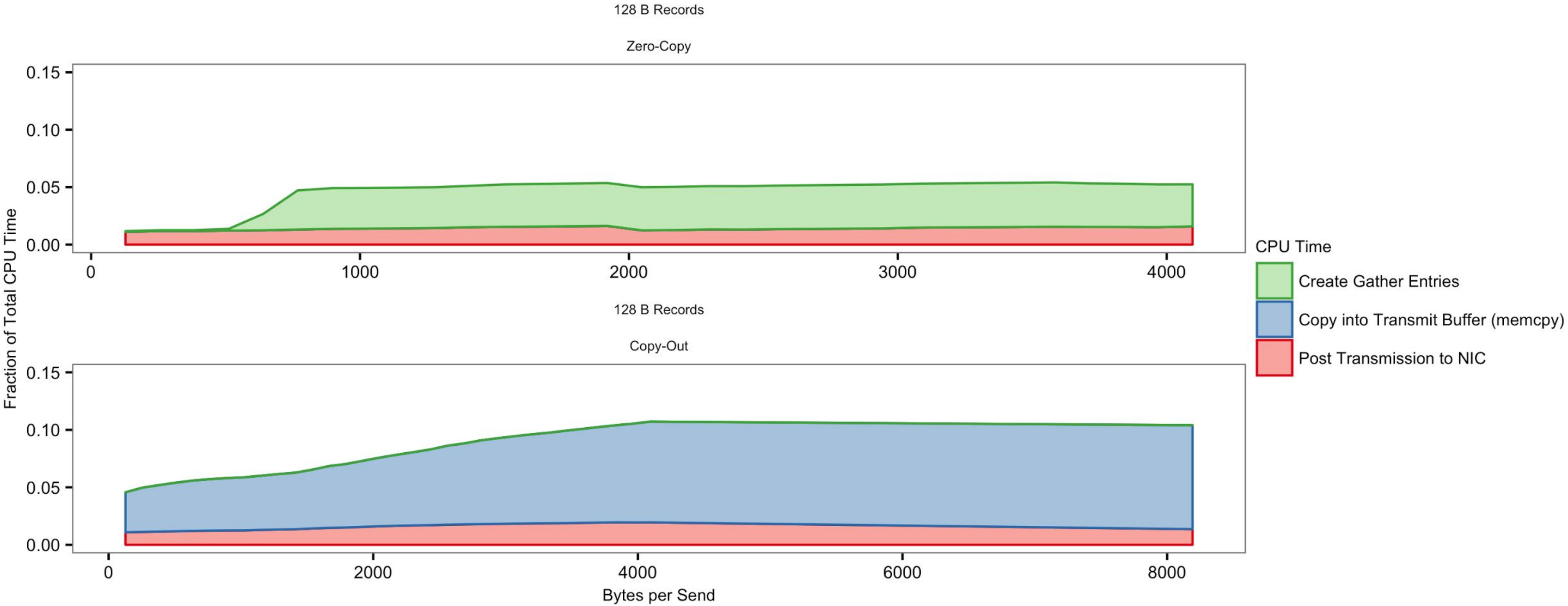
QnA

Source:

To Copy or Not to Copy: Making In-Memory Databases Fast on Modern NICs:

- A.Kesavan, R. Ricci, R. Stutsman

CPU Overheads


128 B Records
Zero-Copy
128 B Records
Copy-Out
Bytes per Send
Fraction of Total CPU Time
CPU Time
Create Gather Entries
Copy into Transmit Buffer (memcpy)
Post Transmission to NIC